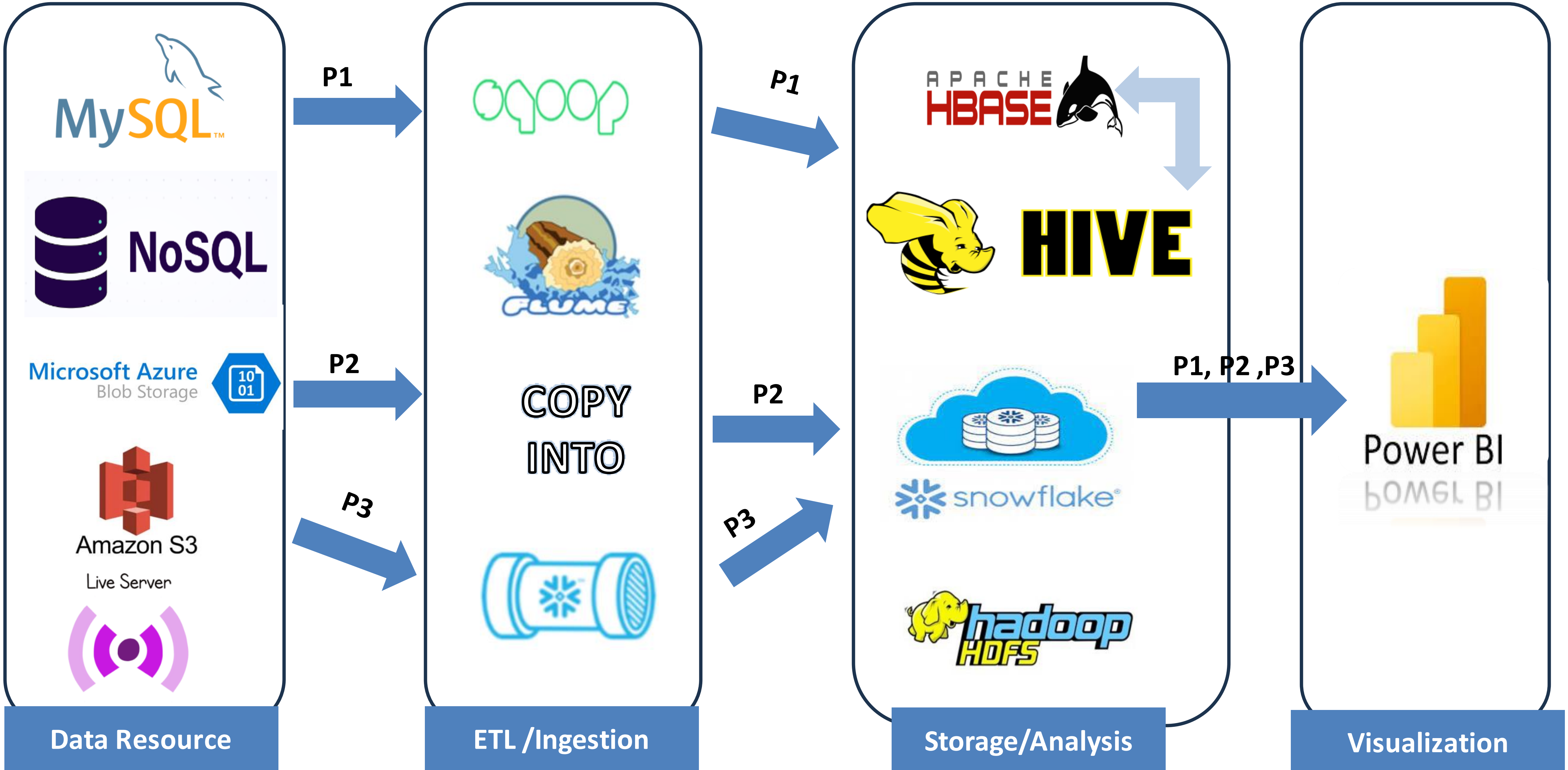
Olympic Data Engineering Solution

Objectives

✓Ingest data from various sources into the data ecosystem.
✓Transform and store data efficiently for analysis.
✓Enable data analysis and reporting capabilities for Olympic Games insights.
✓Ensure data security and compliance with relevant regulations.

Data Engineering Architecture Diagram



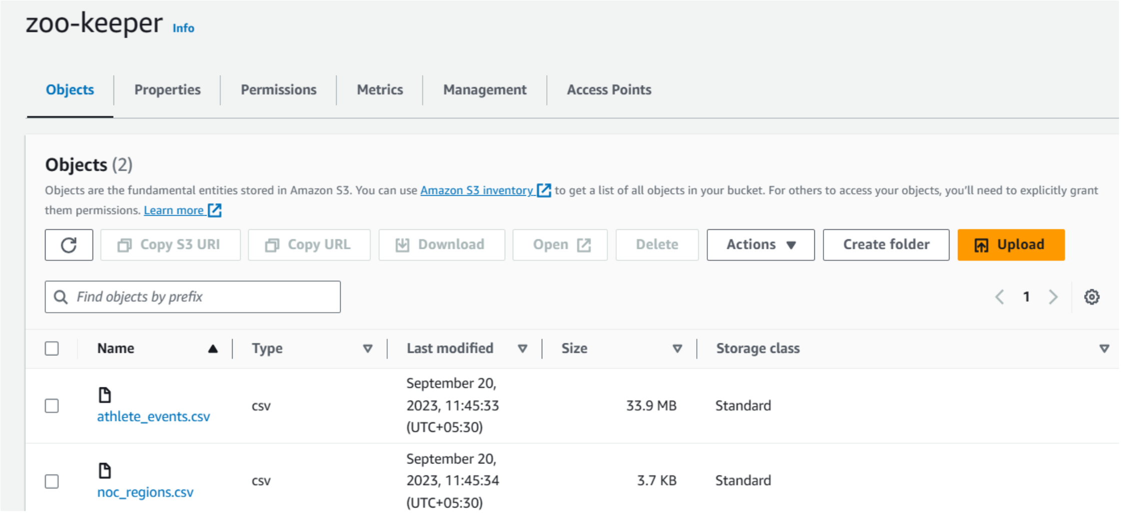
Data Source

Microsoft Azure
Blob Storage

MySQL™

Amazon S3

Amazon S3 Bucket

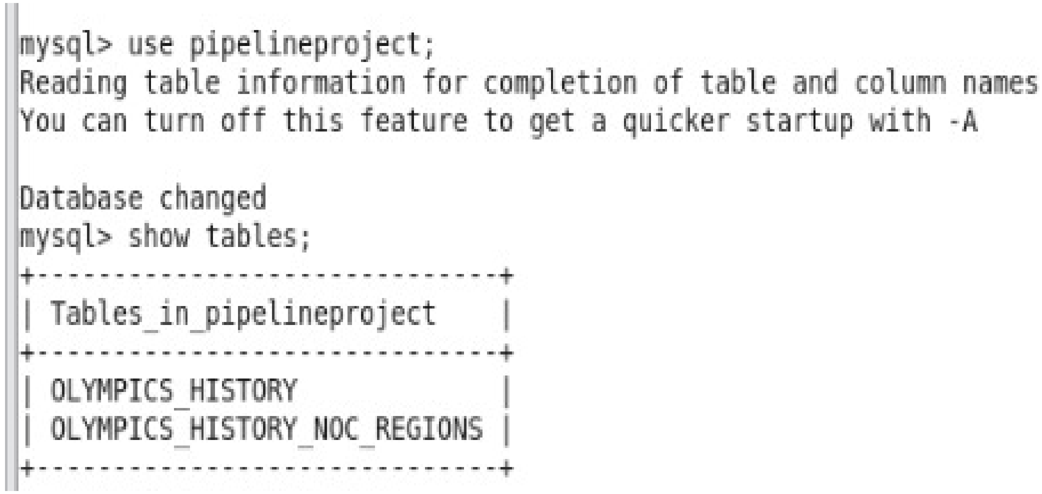


Objects (2)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

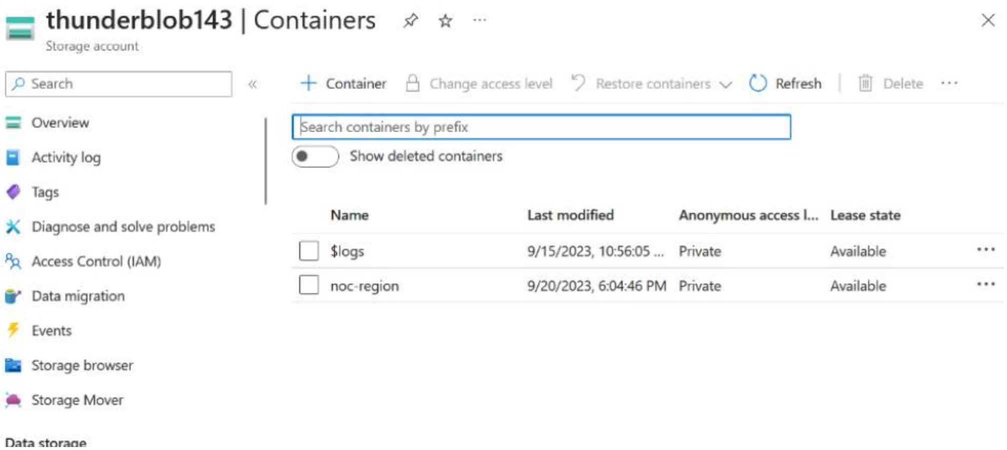
□	Name	Type	Last modified	Size	Storage class
□	athlete_events.csv	csv	September 20, 2023, 11:45:33 (UTC+05:30)	33.9 MB	Standard
□	noc_regions.csv	csv	September 20, 2023, 11:45:34 (UTC+05:30)	3.7 KB	Standard

MYSQL DB



```
mysql> use pipelineproject;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-----------------------------+
| Tables_in_pipelineproject   |
+-----------------------------+
| OLYMPICS_HISTORY            |
| OLYMPICS_HISTORY_NOC_REGIONS |
+-----------------------------+
```

Azure Blob Container



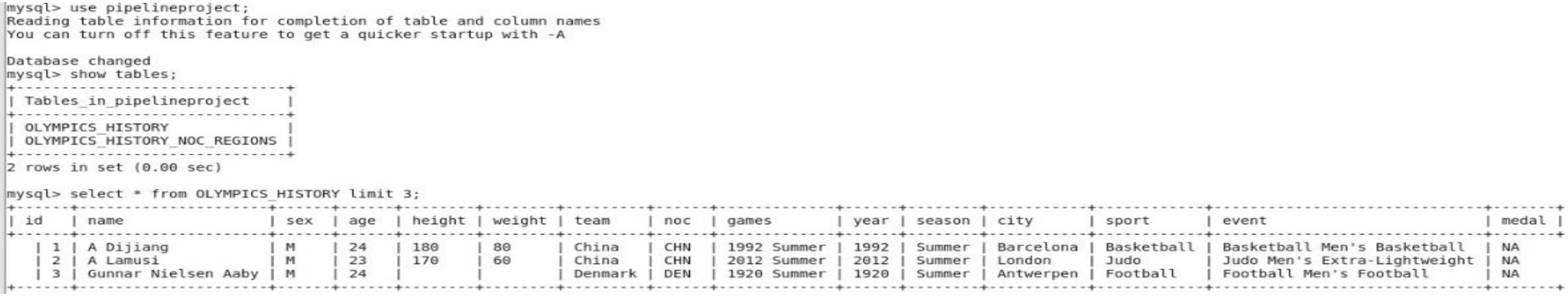
	Name	Last modified	Anonymous access l...	Lease state
□	$logs	9/15/2023, 10:56:05 ...	Private	Available
□	noc-region	9/20/2023, 6:04:46 PM	Private	Available

Ingestion/ETL

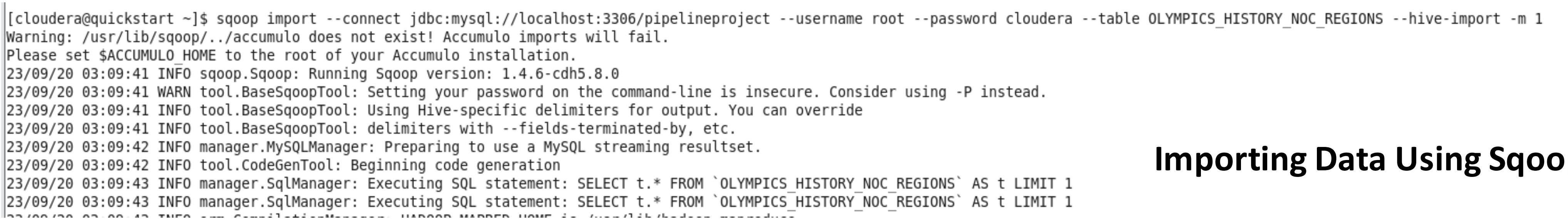
APACHE SQOOP

COPY INTO

MYSQL to HIVE (SQOOP)

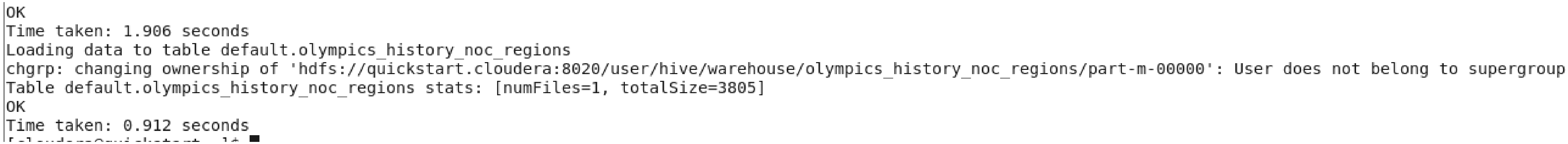


```
mysql> use pipelineproject;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-------------------------------+
| Tables_in_pipelineproject     |
+-------------------------------+
| OLYMPICS_HISTORY              |
| OLYMPICS_HISTORY_NOC_REGIONS  |
+-------------------------------+
2 rows in set (0.00 sec)

mysql> select * from OLYMPICS_HISTORY limit 3;
+------+----------------------+-----+-----+--------+--------+---------+-----+-------------+------+--------+-----------+------------+-------------------------------+-------+
| id   | name                 | sex | age | height | weight | team    | noc | games       | year | season | city      | sport      | event                         | medal |
+------+----------------------+-----+-----+--------+--------+---------+-----+-------------+------+--------+-----------+------------+-------------------------------+-------+
|    1 | A Dijiang            | M   | 24  | 180    | 80     | China   | CHN | 1992 Summer | 1992 | Summer | Barcelona | Basketball | Basketball Men's Basketball   | NA    |
|    2 | A Lamusi             | M   | 23  | 170    | 60     | China   | CHN | 2012 Summer | 2012 | Summer | London    | Judo       | Judo Men's Extra-Lightweight  | NA    |
|    3 | Gunnar Nielsen Aaby  | M   | 24  |        |        | Denmark | DEN | 1920 Summer | 1920 | Summer | Antwerpen | Football   | Football Men's Football       | NA    |
+------+----------------------+-----+-----+--------+--------+---------+-----+-------------+------+--------+-----------+------------+-------------------------------+-------+
```



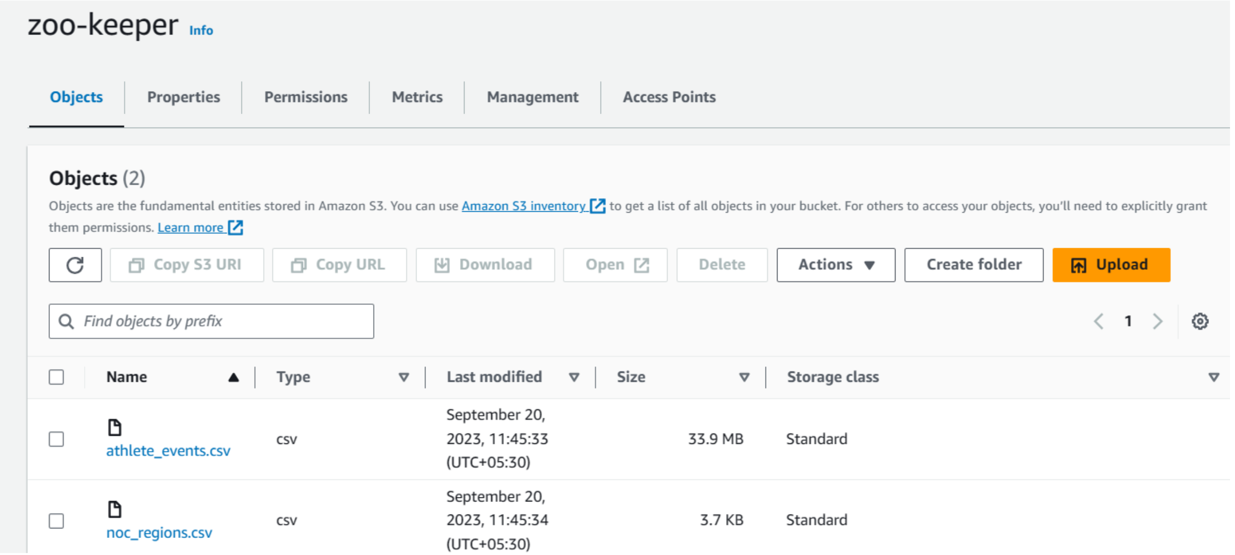
```
[cloudera@quickstart ~]$ sqoop import --connect jdbc:mysql://localhost:3306/pipelineproject --username root --password cloudera --table OLYMPICS_HISTORY_NOC_REGIONS --hive-import -m 1
Warning: /usr/lib/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
23/09/20 03:09:41 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.8.0
23/09/20 03:09:41 WARN tool.BaseSqoopTool: Setting your password on the command-line is insecure. Consider using -P instead.
23/09/20 03:09:41 INFO tool.BaseSqoopTool: Using Hive-specific delimiters for output. You can override
23/09/20 03:09:41 INFO tool.BaseSqoopTool: delimiters with --fields-terminated-by, etc.
23/09/20 03:09:42 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.
23/09/20 03:09:42 INFO tool.CodeGenTool: Beginning code generation
23/09/20 03:09:43 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `OLYMPICS_HISTORY_NOC_REGIONS` AS t LIMIT 1
23/09/20 03:09:43 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `OLYMPICS_HISTORY_NOC_REGIONS` AS t LIMIT 1
```

Importing Data Using Sqoop



```
OK
Time taken: 1.906 seconds
Loading data to table default.olympics_history_noc_regions
chgrp: changing ownership of 'hdfs://quickstart.cloudera:8020/user/hive/warehouse/olympics_history_noc_regions/part-m-00000': User does not belong to supergroup
Table default.olympics_history_noc_regions stats: [numFiles=1, totalSize=3805]
OK
Time taken: 0.912 seconds
```

Amazon S3 to Snowflake (Snow pipe)



zoo-keeper Info

Objects | Properties | Permissions | Metrics | Management | Access Points

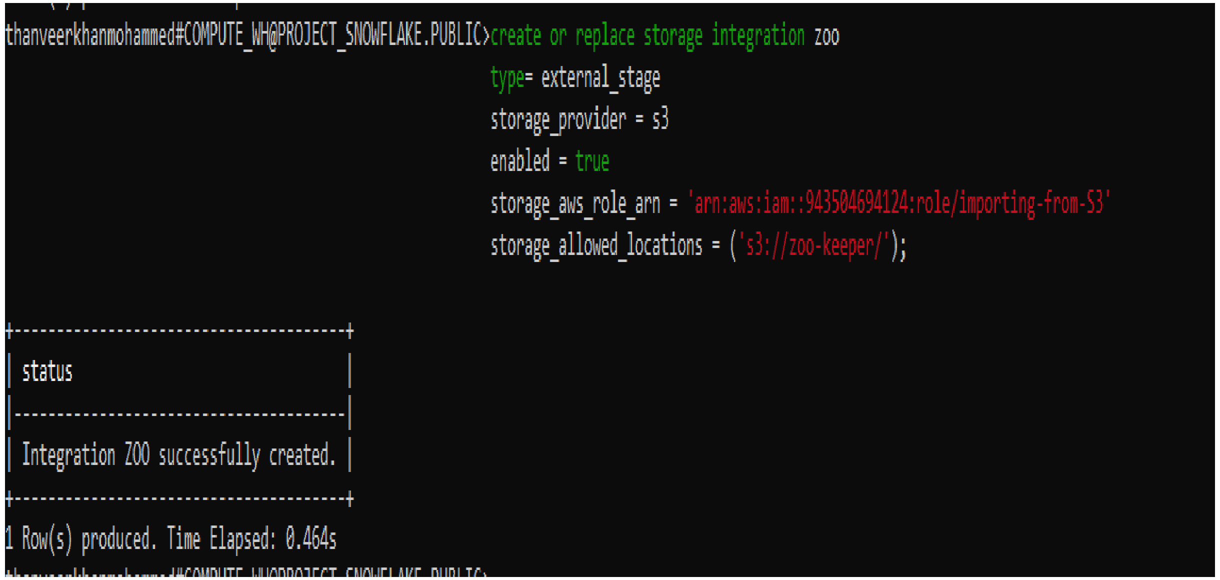
Objects (2)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

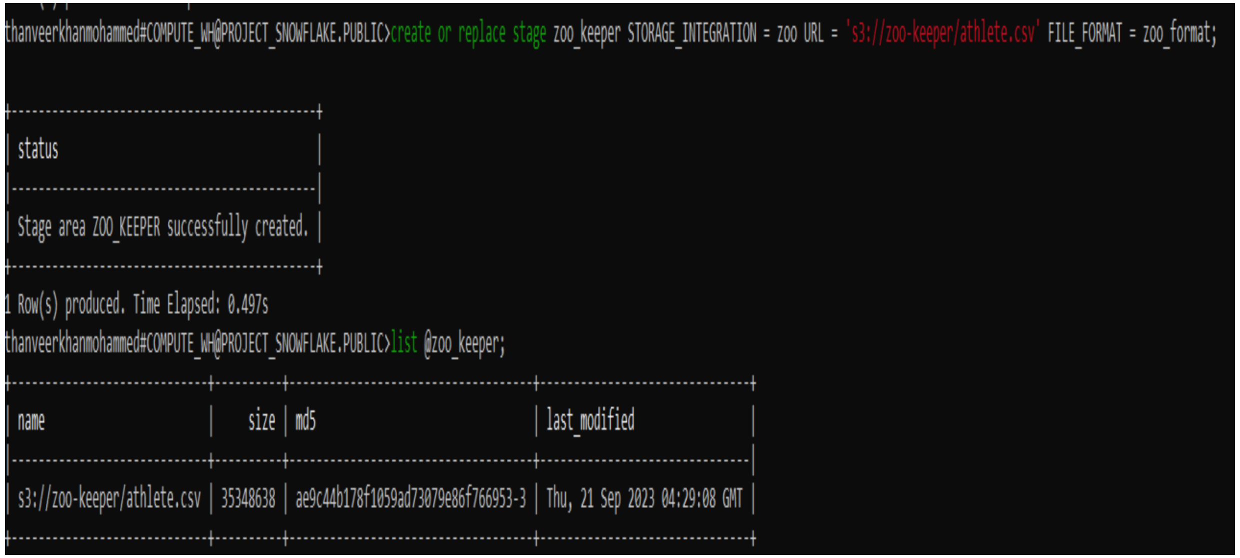
Copy S3 URI | Copy URL | Download | Open | Delete | Actions ▼ | Create folder | Upload

Find objects by prefix

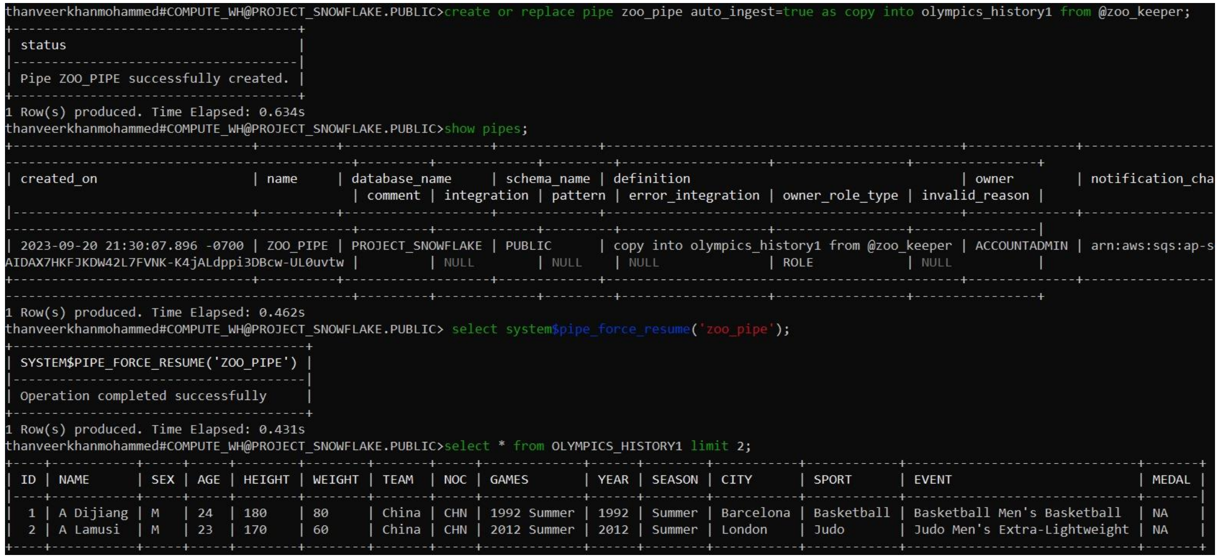
□	Name ▲	Type	Last modified	Size	Storage class
□	athlete_events.csv	csv	September 20, 2023, 11:45:33 (UTC+05:30)	33.9 MB	Standard
□	noc_regions.csv	csv	September 20, 2023, 11:45:34 (UTC+05:30)	3.7 KB	Standard



```
thanveerkhanmohammed#COMPUTE_WH@PROJECT_SNOWFLAKE.PUBLIC>create or replace storage integration zoo
                                                         type= external_stage
                                                         storage_provider = s3
                                                         enabled = true
                                                         storage_aws_role_arn = 'arn:aws:iam::943504694124:role/importing-from-S3'
                                                         storage_allowed_locations = ('s3://zoo-keeper/');

+----------------------------------+
| status                           |
|----------------------------------|
| Integration ZOO successfully created. |
+----------------------------------+
1 Row(s) produced. Time Elapsed: 0.464s
thanveerkhanmohammed#COMPUTE_WH@PROJECT_SNOWFLAKE.PUBLIC>
```

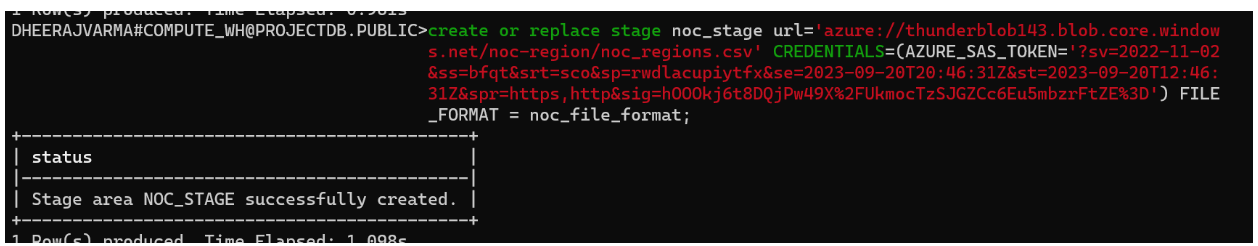


```
thanveerkhanmohammed#COMPUTE_WH@PROJECT_SNOWFLAKE.PUBLIC>create or replace stage zoo_keeper STORAGE_INTEGRATION = zoo URL = 's3://zoo-keeper/athlete.csv' FILE_FORMAT = zoo_format;

+-----------------------------------+
| status                            |
|-----------------------------------|
| Stage area ZOO_KEEPER successfully created. |
+-----------------------------------+
1 Row(s) produced. Time Elapsed: 0.497s
thanveerkhanmohammed#COMPUTE_WH@PROJECT_SNOWFLAKE.PUBLIC>list @zoo_keeper;
+---------------------------+----------+----------------------------------+------------------------------+
| name                      |     size | md5                              | last_modified                |
|---------------------------+----------+----------------------------------+------------------------------|
| s3://zoo-keeper/athlete.csv | 35348638 | ae9c44b178f1059ad73079e86f766953-3 | Thu, 21 Sep 2023 04:29:08 GMT |
+---------------------------+----------+----------------------------------+------------------------------+
```

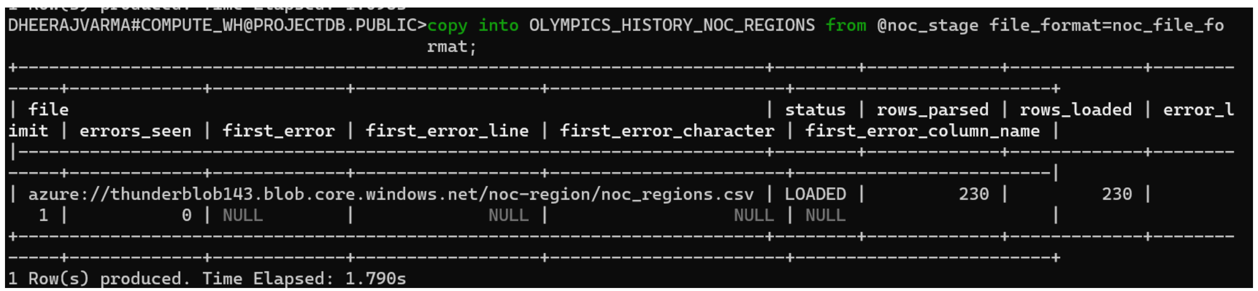


```
thanveerkhanmohammed#COMPUTE_WH@PROJECT_SNOWFLAKE.PUBLIC>create or replace pipe zoo_pipe auto_ingest=true as copy into olympics_history1 from @zoo_keeper;
+-----------------------------------+
| status                            |
|-----------------------------------|
| Pipe ZOO_PIPE successfully created. |
+-----------------------------------+
1 Row(s) produced. Time Elapsed: 0.634s
thanveerkhanmohammed#COMPUTE_WH@PROJECT_SNOWFLAKE.PUBLIC>show pipes;
+-------------------------------+----------+-------------------+-------------+------------------------------------------+-------------+--------------------
-----------------------------------------+---------+-------------+---------+-------------------+-----------------+----------------+
| created_on                    | name     | database_name     | schema_name | definition                               | owner       | notification_cha
                                         | comment | integration | pattern | error_integration | owner_role_type | invalid_reason |
|-------------------------------+----------+-------------------+-------------+------------------------------------------+-------------+--------------------
-----------------------------------------+---------+-------------+---------+-------------------+-----------------+----------------|
| 2023-09-20 21:30:07.896 -0700 | ZOO_PIPE | PROJECT_SNOWFLAKE | PUBLIC      | copy into olympics_history1 from @zoo_keeper | ACCOUNTADMIN | arn:aws:sqs:ap-s
AIDAX7HKFJKDW42L7FVNK-K4jALdppi3DBcw-UL0uvtw |         | NULL        | NULL    | NULL              | ROLE            | NULL           |
+-------------------------------+----------+-------------------+-------------+------------------------------------------+-------------+--------------------
-----------------------------------------+---------+-------------+---------+-------------------+-----------------+----------------+
1 Row(s) produced. Time Elapsed: 0.462s
thanveerkhanmohammed#COMPUTE_WH@PROJECT_SNOWFLAKE.PUBLIC> select system$pipe_force_resume('zoo_pipe');
+-----------------------------------+
| SYSTEM$PIPE_FORCE_RESUME('ZOO_PIPE') |
|-----------------------------------|
| Operation completed successfully  |
+-----------------------------------+
1 Row(s) produced. Time Elapsed: 0.431s
thanveerkhanmohammed#COMPUTE_WH@PROJECT_SNOWFLAKE.PUBLIC>select * from OLYMPICS_HISTORY1 limit 2;
+----+-----------+-----+-----+--------+--------+-------+-----+-------------+------+--------+-----------+------------+-------------------------------+-------+
| ID | NAME      | SEX | AGE | HEIGHT | WEIGHT | TEAM  | NOC | GAMES       | YEAR | SEASON | CITY      | SPORT      | EVENT                         | MEDAL |
|----+-----------+-----+-----+--------+--------+-------+-----+-------------+------+--------+-----------+------------+-------------------------------+-------|
|  1 | A Dijiang | M   | 24  | 180    | 80     | China | CHN | 1992 Summer | 1992 | Summer | Barcelona | Basketball | Basketball Men's Basketball   | NA    |
|  2 | A Lamusi  | M   | 23  | 170    | 60     | China | CHN | 2012 Summer | 2012 | Summer | London    | Judo       | Judo Men's Extra-Lightweight  | NA    |
+----+-----------+-----+-----+--------+--------+-------+-----+-------------+------+--------+-----------+------------+-------------------------------+-------+
```

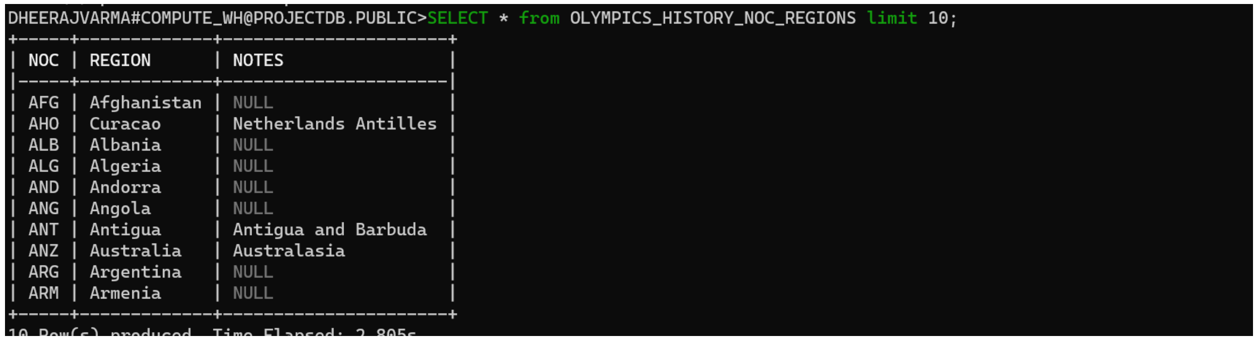
Azure Blob to Snowflake (COPY INTO)



```
DHEERAJVARMA#COMPUTE_WH@PROJECTDB.PUBLIC>create or replace stage noc_stage url='azure://thunderblob143.blob.core.window
                                         s.net/noc-region/noc_regions.csv' CREDENTIALS=(AZURE_SAS_TOKEN='?sv=2022-11-02
                                         &ss=bfqt&srt=sco&sp=rwdlacupiytfx&se=2023-09-20T20:46:31Z&st=2023-09-20T12:46:
                                         31Z&spr=https,http&sig=hOOOkj6t8DQjPw49X%2FUkmocTzSJGZCc6Eu5mbzrFtZE%3D') FILE
                                         _FORMAT = noc_file_format;
+--------------------------------------------+
| status                                     |
|--------------------------------------------|
| Stage area NOC_STAGE successfully created. |
+--------------------------------------------+
1 Row(s) produced. Time Elapsed: 1.098s
```



```
DHEERAJVARMA#COMPUTE_WH@PROJECTDB.PUBLIC>copy into OLYMPICS_HISTORY_NOC_REGIONS from @noc_stage file_format=noc_file_fo
                                         rmat;
+-------------------------------------------------------------------------+--------+-------------+-------------+--------
-----+-------------+-------------+------------------+-----------------------+-------------------------+
| file                                                                    | status | rows_parsed | rows_loaded | error_l
imit | errors_seen | first_error | first_error_line | first_error_character | first_error_column_name |
|-------------------------------------------------------------------------+--------+-------------+-------------+--------
-----+-------------+-------------+------------------+-----------------------+-------------------------|
| azure://thunderblob143.blob.core.windows.net/noc-region/noc_regions.csv | LOADED |         230 |         230 |
   1 |           0 | NULL        |             NULL |                  NULL | NULL                    |
+-------------------------------------------------------------------------+--------+-------------+-------------+--------
-----+-------------+-------------+------------------+-----------------------+-------------------------+
1 Row(s) produced. Time Elapsed: 1.790s
```

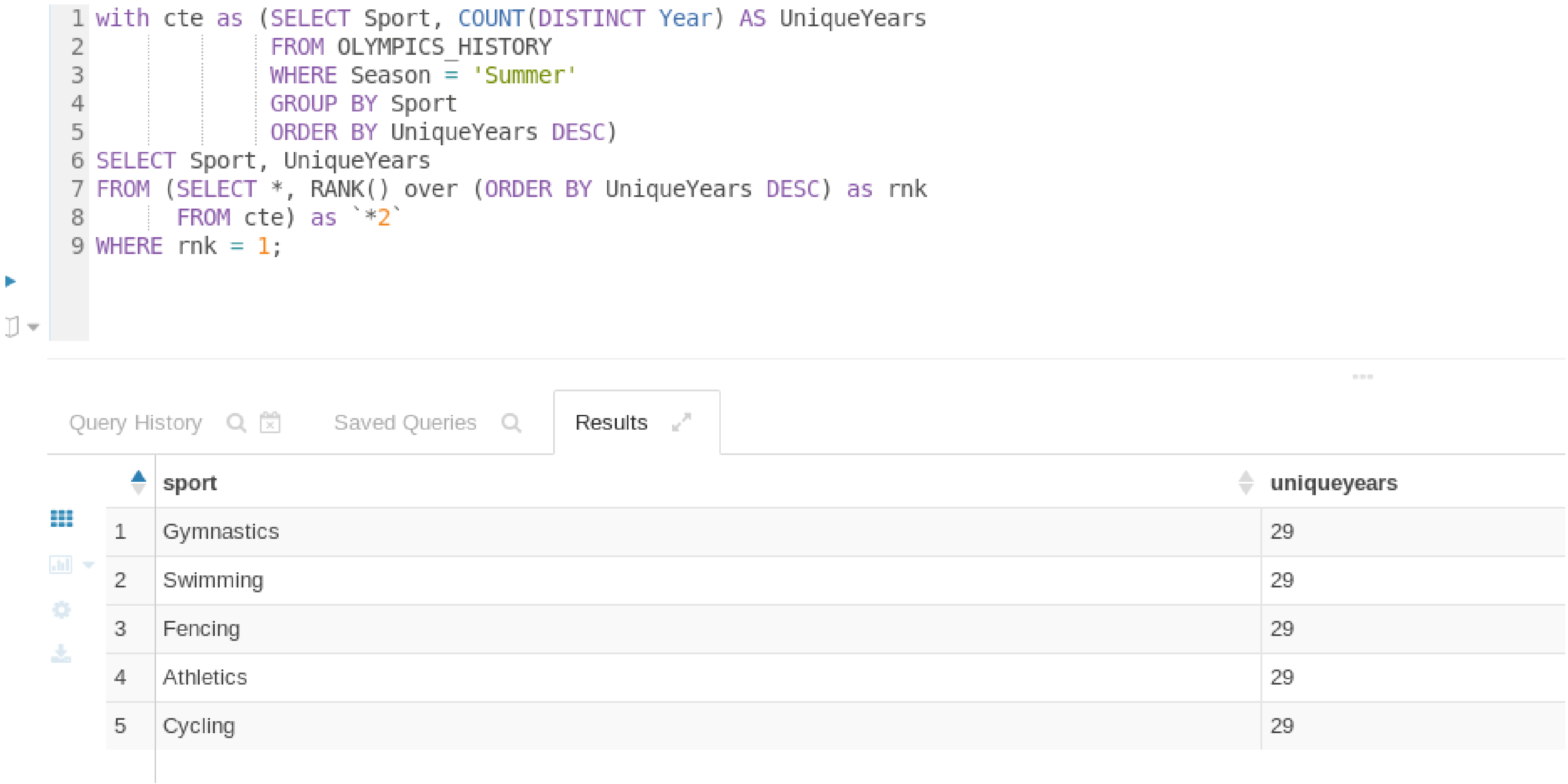


```
DHEERAJVARMA#COMPUTE_WH@PROJECTDB.PUBLIC>SELECT * from OLYMPICS_HISTORY_NOC_REGIONS limit 10;
+-----+-------------+----------------------+
| NOC | REGION      | NOTES                |
|-----+-------------+----------------------|
| AFG | Afghanistan | NULL                 |
| AHO | Curacao     | Netherlands Antilles |
| ALB | Albania     | NULL                 |
| ALG | Algeria     | NULL                 |
| AND | Andorra     | NULL                 |
| ANG | Angola      | NULL                 |
| ANT | Antigua     | Antigua and Barbuda  |
| ANZ | Australia   | Australasia          |
| ARG | Argentina   | NULL                 |
| ARM | Armenia     | NULL                 |
+-----+-------------+----------------------+
10 Row(s) produced. Time Elapsed: 2.805s
```

Storage and Analysis

HIVE ANALYSIS

Problem Statement : Identify the sport which was played in all summer Olympics.

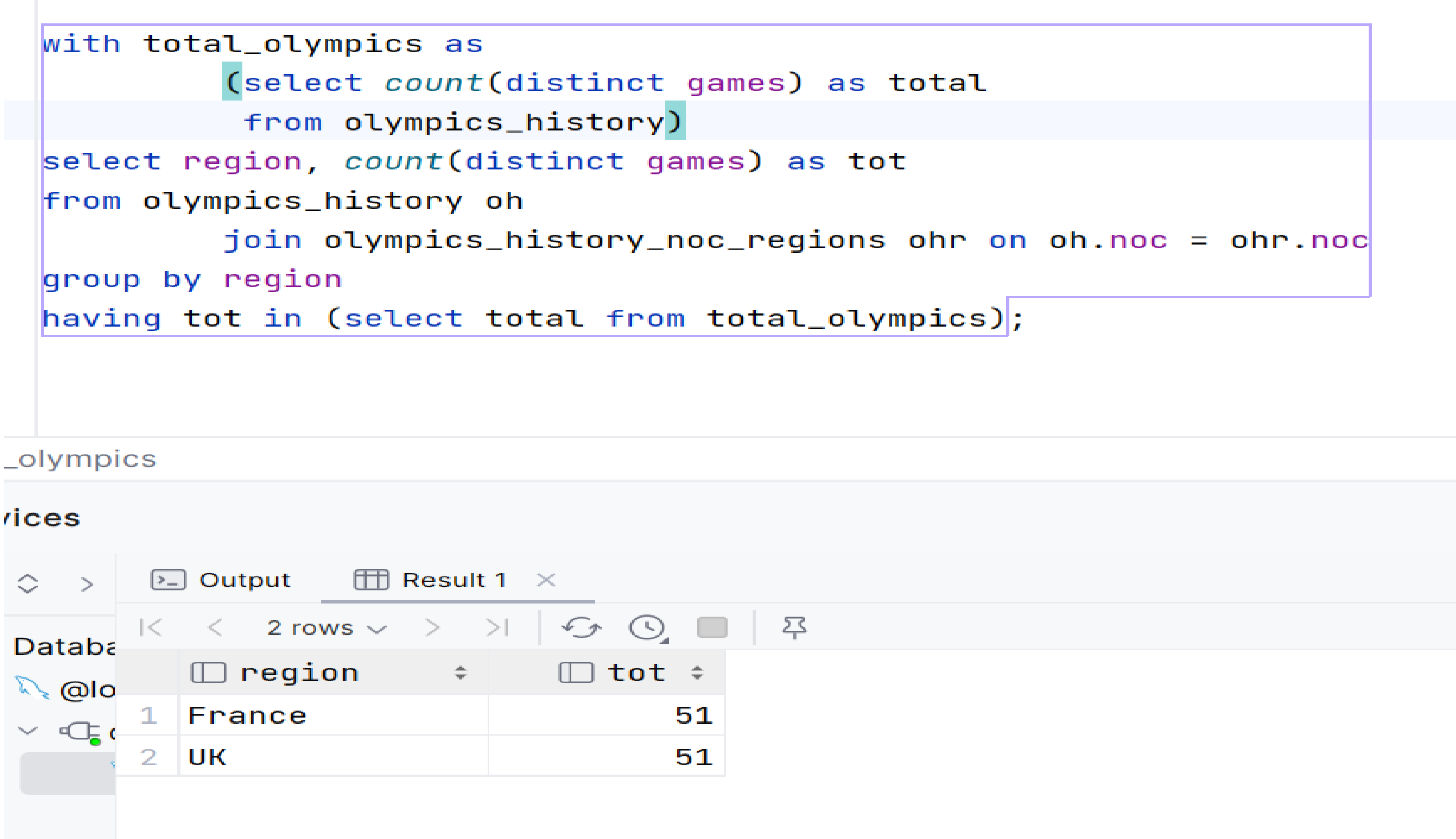


```
with cte as (SELECT Sport, COUNT(DISTINCT Year) AS UniqueYears
             FROM OLYMPICS_HISTORY
             WHERE Season = 'Summer'
             GROUP BY Sport
             ORDER BY UniqueYears DESC)
SELECT Sport, UniqueYears
FROM (SELECT *, RANK() over (ORDER BY UniqueYears DESC) as rnk
      FROM cte) as `*2`
WHERE rnk = 1;
```

	sport	uniqueyears
1	Gymnastics	29
2	Swimming	29
3	Fencing	29
4	Athletics	29
5	Cycling	29

HIVE ANALYSIS

Problem Statement : In which country has participated in all Olympic games.

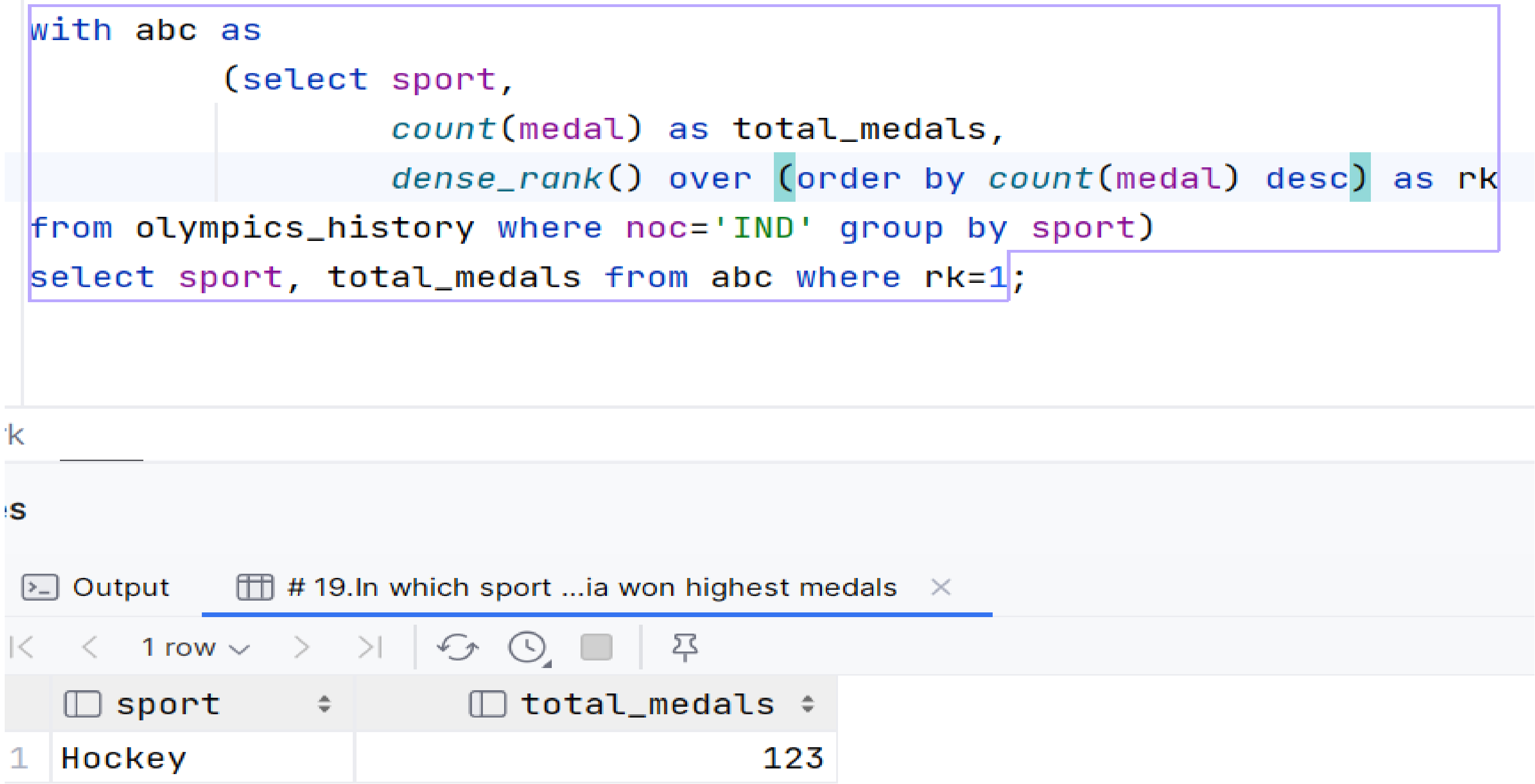


```
with total_olympics as
         (select count(distinct games) as total
          from olympics_history)
select region, count(distinct games) as tot
from olympics_history oh
         join olympics_history_noc_regions ohr on oh.noc = ohr.noc
group by region
having tot in (select total from total_olympics);
```

	region	tot
1	France	51
2	UK	51

APACHE HBASE ANALYSIS

Problem Statement : In which sport India won highest medals

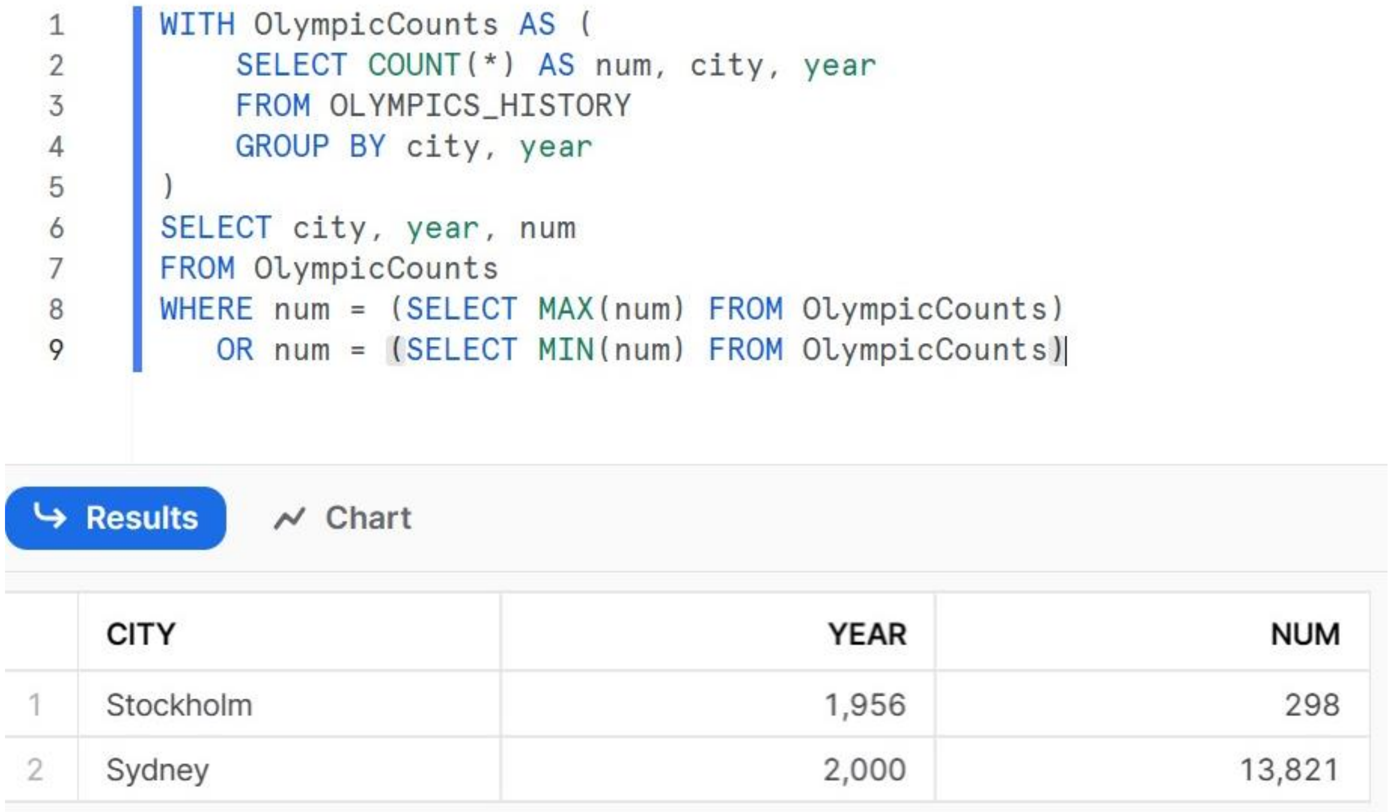


```
with abc as
         (select sport,
                 count(medal) as total_medals,
                 dense_rank() over (order by count(medal) desc) as rk
from olympics_history where noc='IND' group by sport)
select sport, total_medals from abc where rk=1;
```

	sport	total_medals
1	Hockey	123

ANALYSIS

Problem Statement : Which year saw the highest and lowest no of countries participating in Olympics



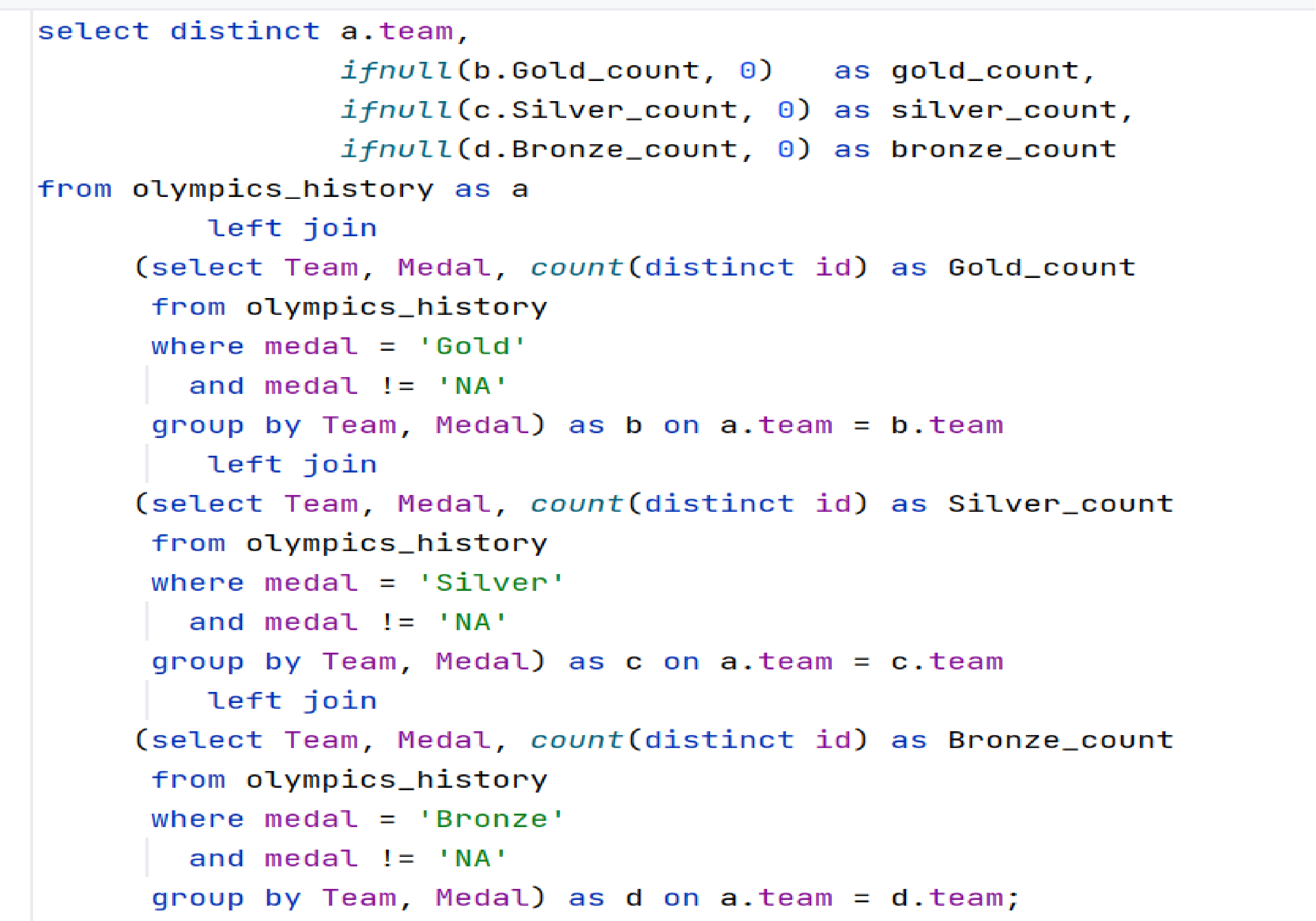
```sql
WITH OlympicCounts AS (
    SELECT COUNT(*) AS num, city, year
    FROM OLYMPICS_HISTORY
    GROUP BY city, year
)
SELECT city, year, num
FROM OlympicCounts
WHERE num = (SELECT MAX(num) FROM OlympicCounts)
   OR num = (SELECT MIN(num) FROM OlympicCounts)
```

	CITY	YEAR	NUM
1	Stockholm	1,956	298
2	Sydney	2,000	13,821

snowflake

ANALYSIS

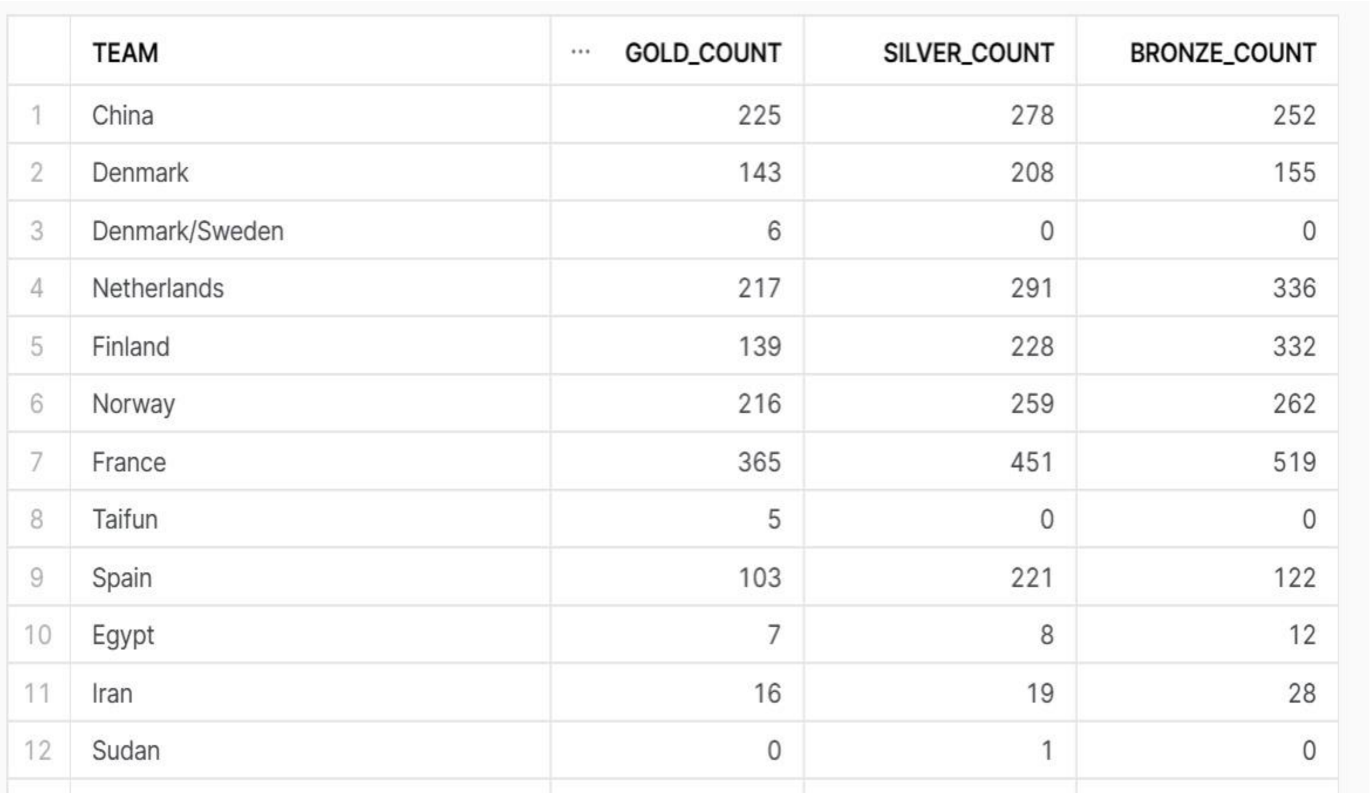
Problem Statement : List down total gold, silver and bronze medals won by each country



```
select distinct a.team,
                ifnull(b.Gold_count, 0)   as gold_count,
                ifnull(c.Silver_count, 0) as silver_count,
                ifnull(d.Bronze_count, 0) as bronze_count
from olympics_history as a
         left join
     (select Team, Medal, count(distinct id) as Gold_count
      from olympics_history
      where medal = 'Gold'
        and medal != 'NA'
      group by Team, Medal) as b on a.team = b.team
         left join
     (select Team, Medal, count(distinct id) as Silver_count
      from olympics_history
      where medal = 'Silver'
        and medal != 'NA'
      group by Team, Medal) as c on a.team = c.team
         left join
     (select Team, Medal, count(distinct id) as Bronze_count
      from olympics_history
      where medal = 'Bronze'
        and medal != 'NA'
      group by Team, Medal) as d on a.team = d.team;
```

ANALYSIS

OUTPUT

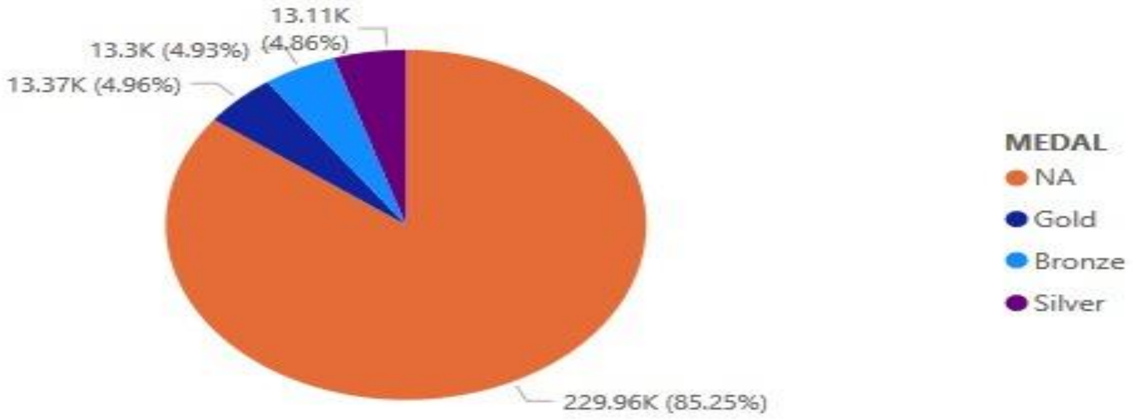


	TEAM	GOLD_COUNT	SILVER_COUNT	BRONZE_COUNT
1	China	225	278	252
2	Denmark	143	208	155
3	Denmark/Sweden	6	0	0
4	Netherlands	217	291	336
5	Finland	139	228	332
6	Norway	216	259	262
7	France	365	451	519
8	Taifun	5	0	0
9	Spain	103	221	122
10	Egypt	7	8	12
11	Iran	16	19	28
12	Sudan	0	1	0

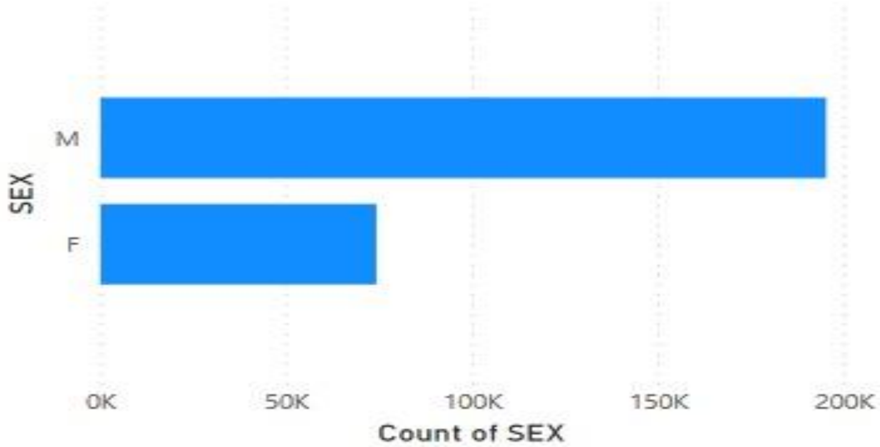
Power BI

Visualization

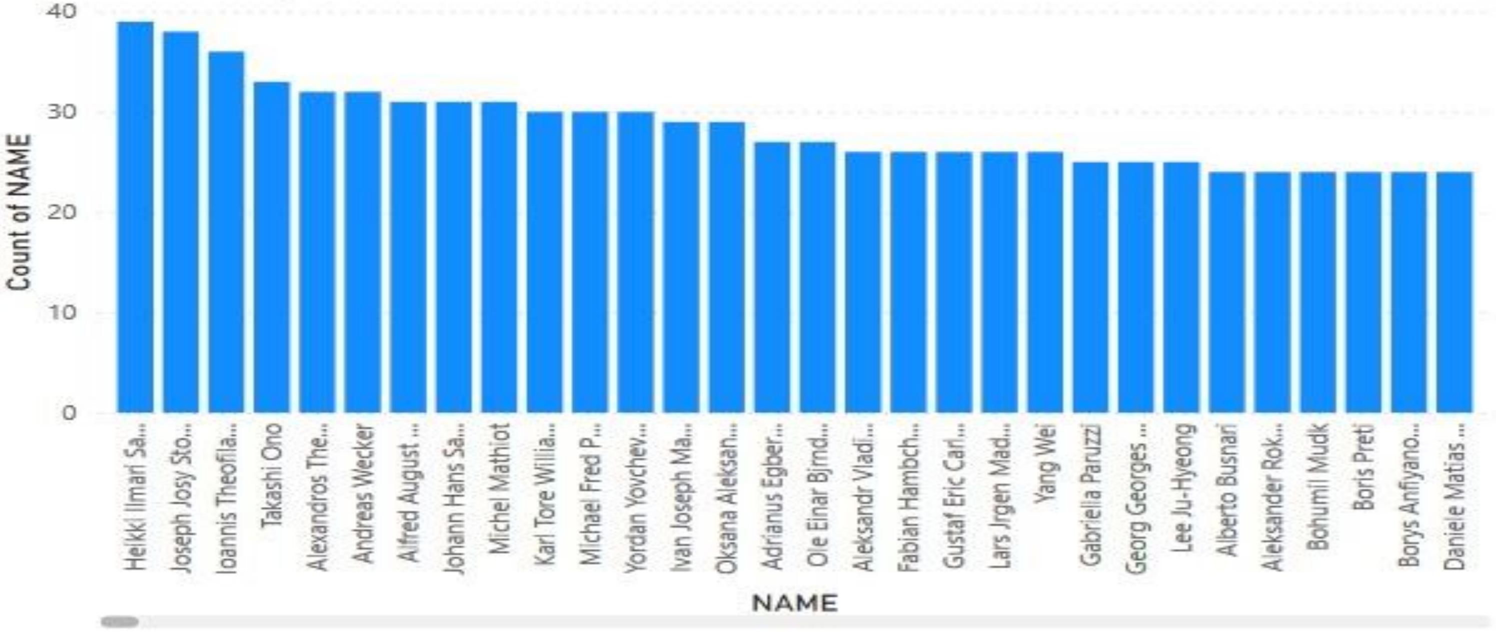
Count of MEDAL by MEDAL



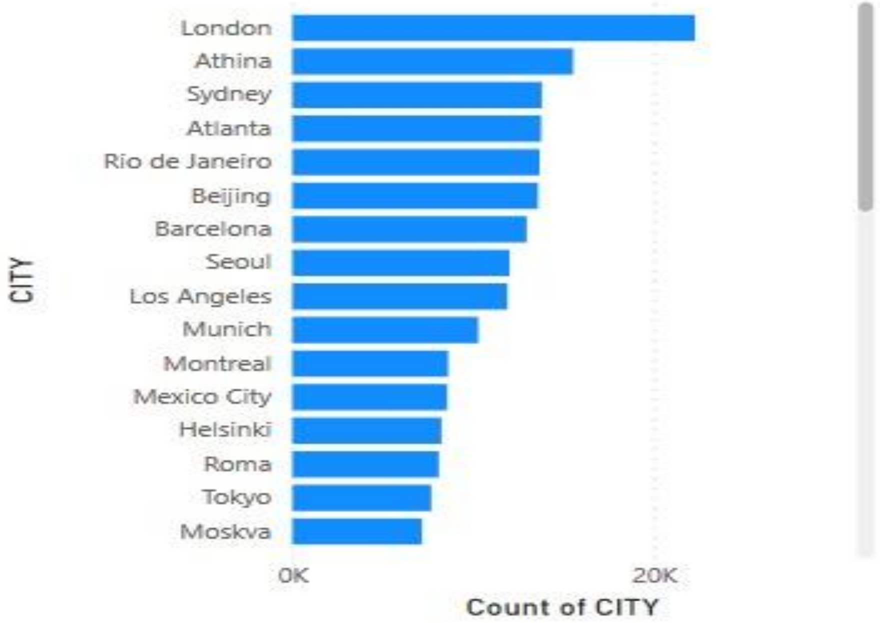
Count of SEX by SEX



Count of NAME by NAME



Count of CITY by CITY


London
Athina
Sydney
Atlanta
Rio de Janeiro
Beijing
Barcelona
Seoul
Los Angeles
Munich
Montreal
Mexico City
Helsinki
Roma
Tokyo
Moskva
CITY
0K
20K
Count of CITY